Preliminary Analysis of Torque-Free Motion and Gravity Gradient on a Spacecraft

Cameron Davis[1]

Embry-Riddle Aeronautical University, Daytona Beach, Florida, 32114, USA

Charles Keer[2]

Embry-Riddle Aeronautical University, Daytona Beach, Florida, 32114, USA

Martin Kovachev[3]

Embry-Riddle Aeronautical University, Daytona Beach, Florida, 32114, USA

The stability of rigid bodies strongly correlates with the inertial characteristics of the body in question. In this analysis, an axisymmetric rigid body satellite was simulated with different inertial properties to investigate the stability of each case. It was found that by varying the size of the inertial moments of the A, B and C axis that the rigid body exhibits different rotational and stability properties. Additionally, the effect of the gravitational moments was simulated along with small perturbations to the rigid body.

A. Nomenclature

ω	=	*angular velocity*
$\dot{\omega}$	=	*angular acceleration*
$\ddot{\omega}$	=	*variation of angular acceleration*
q	=	*quaternion*
I	=	*moment of inertia*
H	=	*angular momentum*
α	=	*angular acceleration*
M	=	*total moment*
S	=	*spin rate*
$\dot{q}$	=	*variation of quaternion*
v	=	*number of revolutions*

B. Introduction

Nearly all satellites in orbit require a control system for orientation. By analyzing the inertial characteristics of the satellite in question, the stability of the system can be determined. From this, control systems can more accurately position satellites to a specific attitude. This can be accomplished through small thrusters as well as inertial wheels.

C. Torque Free Motion Analysis

To start the analysis of a rigid body, the kinematic and dynamic equations of motion must be derived for the body in question. The total angular velocity in terms of quaternions can be derived from Equations (1) and (2).

$$\dot{\overline{q}} = \tfrac{1}{2}(q_4\overline{\omega} + \overline{q} \times \overline{\omega}) \qquad (\,1\,)$$

$$\dot{q_4} = -\tfrac{1}{2}\overline{q} \cdot \overline{\omega} \qquad (\,2\,)$$

By breaking up the total quaternion and total angular velocity into its respective components for a Cartesian coordinate system, the kinematic equations in terms of quaternions and angular velocity can be shown in Equations (3)-(7).

$$\dot{\overline{q}} = \tfrac{1}{2}\begin{bmatrix} q_4\omega_x + (q_2\omega_z - q_3\omega_y) \\ q_4\omega_y + (q_3\omega_x - q_1\omega_z) \\ q_4\omega_z + (q_1\omega_y - q_3\omega_x) \end{bmatrix} \qquad (\,3\,)$$

$$\dot{q_1} = \tfrac{1}{2}[q_4 + (q_2\omega_z - q_3\omega_y)] \qquad (\,4\,)$$

$$\dot{q_2} = \tfrac{1}{2}[q_4 + (q_2\omega_z - q_3\omega_y)] \qquad (\,5\,)$$

$$\dot{q_3} = \tfrac{1}{2}[q_4\omega_z + (q_1\omega_y - q_3)] \qquad (\,6\,)$$

$$\dot{q_4} = -\tfrac{1}{2}(q_1 + q_2\omega_y + q_3\omega_z) \qquad (\,7\,)$$

The components of the quaternions can be found from a simple direction cosine matrix for the body in question. Similarly, the total moment of the system can be found by using Equation (8).

$$\overline{M} = \overline{\dot{H}_{rel}} + \overline{\omega} \times \overline{H} \qquad (\,8\,)$$

This is the classical Euler equation of motion. This angular momentum terms in the equation can be broken down in terms of the moments of inertia and the relative angular acceleration as shown in Equation (9). This equation will be the basis for the torque free analysis of the rigid bodies.

$$\overline{M} = I_G\overline{\alpha_{rel}} + \overline{\omega} \times I_G\overline{\omega} \qquad (\,9\;\,)$$

Replacing the variables in Equation (9) with the matrices containing the values for an arbitrary rigid body gives Equation (10). By further breaking this equation down, the dynamic equations of motion in terms of the angular velocity as well as the moments of inertia, and by applying the torque-free condition, the dynamic equations of motion can be found as shown in Equations (10)-(14).

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}\begin{pmatrix} \dot{\omega_x} \\ \dot{\omega_y} \\ \dot{\omega_z} \end{pmatrix} + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ -\omega_z & 0 & \omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}\begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \qquad (\,10\,)$$

$$\dot{\omega_y} = -\frac{1}{2}\omega_z \qquad (11)$$

$$\dot{\omega_z} = -\frac{3}{4}\omega_y \qquad (12)$$

$$I_1\dot{\omega_x} + (I_3 - I_2)\omega_y\omega_z = 0 \qquad (13)$$

$$I_2\dot{\omega_y} = 0 \qquad (14)$$

$$I_3\dot{\omega_z} + (I_2 - I_1)\omega_x\omega_y = 0 \qquad (15)$$

Three cases were simulated with differing moments of inertia: A;C>B, A;C<B, and A<B<C, where A, B, and C correspond to the principle moments of inertia of the rigid body. For the case of A;C>B, the initial angular velocity was (1, 0.1, 5) radians per second. The angular velocity in the Y axis is a small perturbation induced on the rigid body. As shown in Figure 1, the angular velocity about the B axis stayed constant while the A and C axis varied sinusoidally. This shows that the rigid body in this configuration is stable about the B axis. Likewise, this can also be seen on the A;C<B case. The initial angular velocity was also (1, 0.1, 5) radians per second. Like the first case, the 0.1 radians per second in the Y direction is a small, induced perturbation on the rigid body. The rigid body keeps a constant angular velocity of zero about the B axis while the A and C axis vary sinusoidally. All three axis are stable as they return to their original amplitude for each cycle.

For the case where A<B<C, the angular velocity behaves much differently than the previous two cases. In this case, the rigid body appears to exhibit some precession and nutation. However, the motion appears stable, in that the angular velocity amplitude does not decrease or increase on average across the timespan.

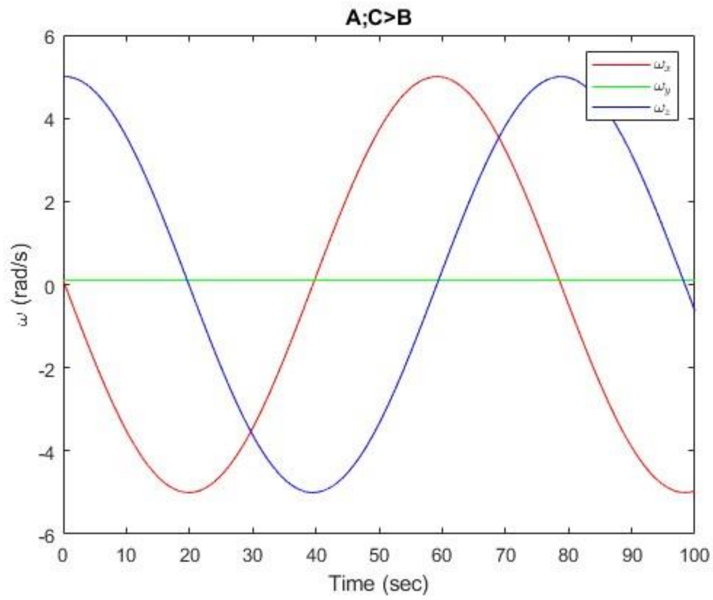

Fig 1. A;C>B. Note the near- zero angular velocity about the B axis.

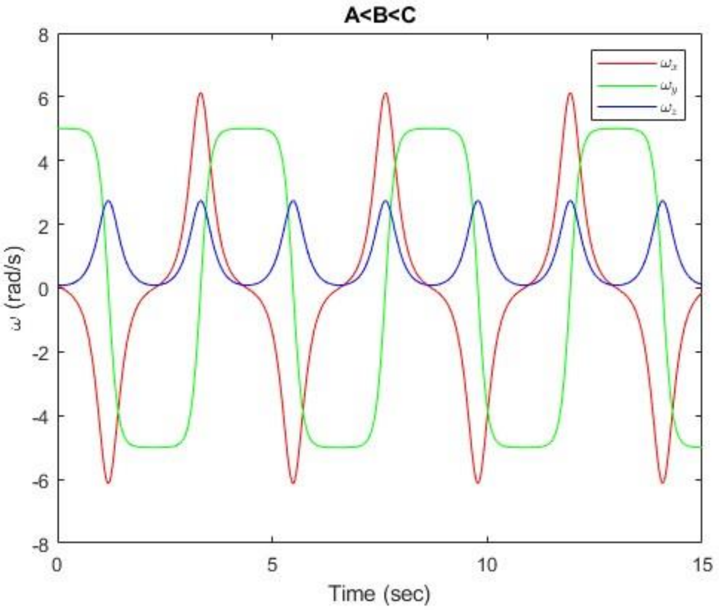

Fig 2. A;C<B. Note the near- zero angular velocity about the B axis.

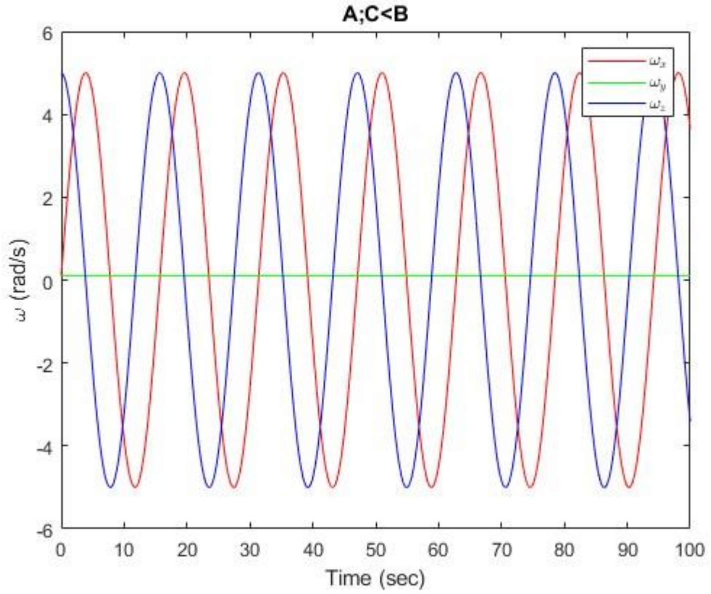

Fig 3. A<B<C. Note the variation of the angular velocity on all three axis.

The custom rigid body that was chosen was a symmetrical "C" shaped spinner. The spinner's inertial characteristics (high moments of inertia due to using Al 6061 as its hypothetical material) can be found in Table 1. The initial angular velocity for this spinner is (0.1, 5, 0.1), where the z axis is a small, induced perturbation. By differentiating Equations (13)-(15), we obtain the differential form of the dynamics equations as shown in Equations (16)-(18).

Table 1. Inertial characteristics of the custom spinner.

Axis	Moment of Inertia (kg m^2)
A	32464.29
B	19350
C	16264.29

$$I_1 \delta \dot{\omega}_x + (I_3 - I_2)\delta\omega_y(\omega_o + \delta\omega_z) = 0 \qquad (\ 16\ )$$

$$I_2 \delta \dot{\omega}_y + (I_1 - I_3)\delta\omega_x(\omega_o + \delta\omega_z) = 0 \qquad (\ 27\ )$$

$$I_3 \delta \dot{\omega}_z + (I_2 - I_1)\delta\omega_x\delta\omega_y = 0 \qquad (\ 38\ )$$

Simplifying these equations yields:

$$\delta\ddot{\omega} + \left[\frac{(I_1 - I_3)(I_2 - I_3)}{I_1 I_2}{\omega_o}^2\right] = 0 \qquad (\ 49\ )$$

It was found for this rigid body that k=1.9896. Since k is greater than 1, the body is naturally stable. This is evident in Figure 4, where the angular velocity along the C axis is constant while the A and B axis vary sinusoidally, with no increase or decrease in the overall amplitude.

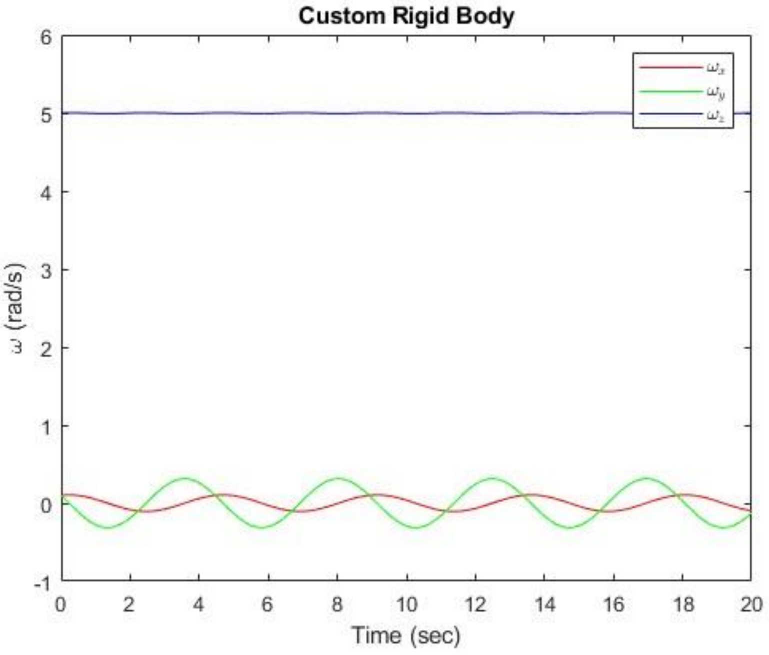

Fig 4. Angular velocities for each axis for the custom "C" spinner.

D. Gravity Gradient Torque Analysis

Starting from Equation (9) from the Torque Free Motion Analysis and considering the spacecraft's angular velocity with respect to the inertially fixed reference frame the kinematic and dynamic equations that govern the satellite's attitude over time. Additionally, the torque due to gravity can be derived as a function of the quaternions starting from Equation (9)

$$\overline{M_G} = \bar{H} + \bar{\omega} \times \bar{H} = I_G \bar{\alpha} + \bar{\Omega} \times I_G \bar{\omega} = \begin{bmatrix} C \\ A \\ A \end{bmatrix} \begin{bmatrix} \dot{\omega_x} \\ \dot{\omega_y} \\ \dot{\omega_z} \end{bmatrix} + \begin{bmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ \omega_x - S & \omega_y & \omega_z \\ C\omega_x & A\omega_y & A\omega_z \end{bmatrix} \qquad (20)$$

$$(M_{Gx} = C\dot{\omega_y} \qquad (21)$$

$$M_{Gy} = A\dot{\omega_y} - A\omega_z(\omega_x - S) + C\omega_x\omega_z \qquad (24)$$

$$M_{Gz} = A\dot{\omega_z} + A\omega_y(\omega_x - S) - C\omega_x\omega_y \qquad (23)$$

$$\dot{\omega_x} = 0 \quad (24)$$

$$\dot{\omega_y} = \frac{12\Omega^2(C-A)[q_2q_3 - q_1q_4][q_1q_2 - q_3q_4] + A\omega_z(\omega_y - S) - C\omega_z\omega_x}{A} \qquad (25)$$

$$\dot{\omega_z} = \frac{6\Omega^2(A-C)[q_1q_2 - q_3q_4][1 - 2{q_1}^2 - 2{q_3}^2] + A\omega_y(\omega_x - S) - C\omega_y\omega_x}{A} \qquad (26)$$

$$\dot{q_1} = \frac{1}{2}\left[q_4(\omega_x - S - \Omega) + q_2\omega_z - q_3\omega_y\right] \qquad (27)$$

$$\dot{q_2} = \frac{1}{2}\left[q_4\omega_y + q_3(\omega_x - S - \Omega) - q_1\omega_z\right] \qquad (28)$$

$$\dot{q_3} = \frac{1}{2}\left[q_4\omega_z + q_1\omega_y - q_2(\omega_x - S - \mathit{\Omega})\right] \qquad (29)$$

$$\dot{q_4} = -\frac{1}{2}\left[q_1(\omega_x - S - \mathit{\Omega}) + q_2\omega_y - q_3\omega_z\right] \; (30)$$

Furthermore, the derived dynamic and kinematic equations of motion from Equations (24)-(30), can be further simplified because the spacecraft's symmetry axis remains normal to the orbit after an initial spin was applied towards the direction normal to the orbit.

$$\dot{\omega_x} = 0 \quad (31)$$

$$\dot{\omega_y} = 12\Omega^2\left(\frac{C}{A} - 1\right)[q_2q_3 - q_1q_4][q_1q_2 + q_3q_4] \qquad (32)$$

$$\dot{\omega_z} = 6\Omega^2\left(1 - \frac{C}{A}\right)[q_1q_2 + q_3q_4][1 - 2{q_1}^2 - 2{q_3}^2] \quad (33)$$

$$\dot{q_1} = \frac{q_4}{2}(\omega_y - S - \Omega) \qquad (34)$$

$$\dot{q_2} = \frac{q_3}{2}(\omega_x - S + \Omega) \qquad (35)$$

$$\dot{q_3} = -\frac{q_2}{2}(\omega_x - S + \Omega) \qquad (36)$$

$$\dot{q_4} = -\frac{q_1}{2}(\omega_x - S - \Omega) \; (37)$$

The results are first nondimensionalized starting from the differential equations, then the results were manipulated such that the independent variable is the number of revolutions instead of the time and the dependent variables are the quaternions and the spacecraft's spin factor and shape factor.

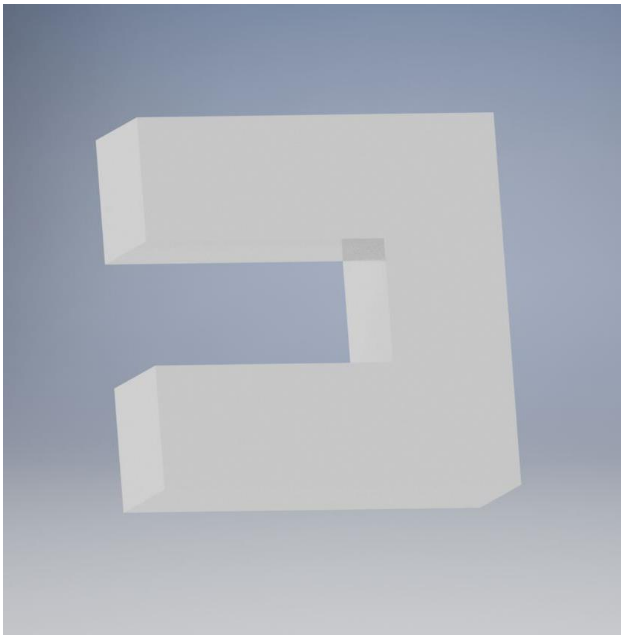

Fig. 5. Render of the custom rigid body.

$$\frac{d_{\omega x}}{d_v} = 0 \; (38)$$

$$\frac{d\omega_y}{d_v} = 2\pi\{-\omega_z S \times [-\omega_x\omega_z + 12(q_2q_3 - q_1q_4)(q_1q_2 + q_3q_4]\} \qquad (39)$$

$$\frac{d\omega_z}{d_v} = 2\pi\{-\omega_y S \times [\omega_y\omega_z - 6(q_1q_2 - q_3q_4)(1 - 2{q_1}^2 - 2{q_3}^2]\} \qquad (40)$$

$$\frac{dq_1}{d_v} = \pi[q_4(\omega_x - S - 1) + q_2\omega_z - q_3\omega_y] \qquad (41)$$

$$\frac{dq_2}{d_v} = \pi[q_4\omega_y + q_3(\omega_x - S + 1) - q_1\omega_z] \qquad (42)$$

$$\frac{dq_3}{d_v} = \pi[q_4\omega_z + q_1\omega_y - q_2(\omega_x - S + 1)] \qquad (43)$$

$$\frac{dq_4}{d_v} = -\pi[q_1(\omega_x - S - 1) + q_2\omega_y - q_3\omega_z] \; (44)$$

With the spacecraft's spin factor and shape factor obtained as well as the dynamic and kinematic equations of motions, the spin rate of the spacecraft can be manipulated such that the solution to the nonlinear differential equations which corresponds to the nominal motion of the spacecraft is constant. For constant nominal motion the variation of quaternions must be equal to zero therefore the following equations can be derived.

$$S = \omega_x - \Omega \qquad (45)$$

$$\dot{q_1} = \frac{q_4}{2}\left(\omega_x - \omega_y + \Omega - \Omega\right) = 0 \qquad (46)$$

$$\dot{q_2} = \frac{q_3}{2}\left(\omega_x - \omega_y + \Omega + \Omega\right) = \Omega q_3 \qquad (47)$$

$$\dot{q_3} = -\frac{q_2}{2}\left(\omega_y - \omega_x + \Omega + \Omega\right) = -\Omega q_2 \qquad (48)$$

$$\dot{q_4} = \frac{q_1}{2}\left(\omega_x - \omega_y + \Omega - \Omega\right) = 0 \qquad (49)$$

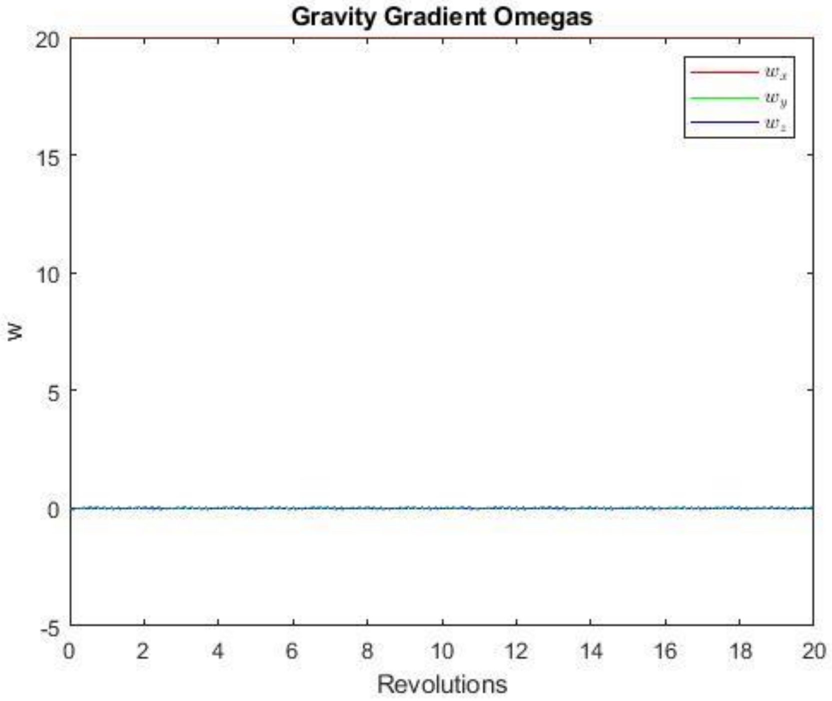

Fig. 6 Gravity Gradient Omegas. Note the constant omegas

Neglecting perturbations, the spacecraft's stability was analyzed with an initial spin direction towards its' C axis and towards its A axis. Several values for the initial spin rate were evaluated, the spacecraft's stability was unaffected, the spacecraft is stable with an initial spin towards its C direction, which is also the direction of which the spacecraft is symmetric.

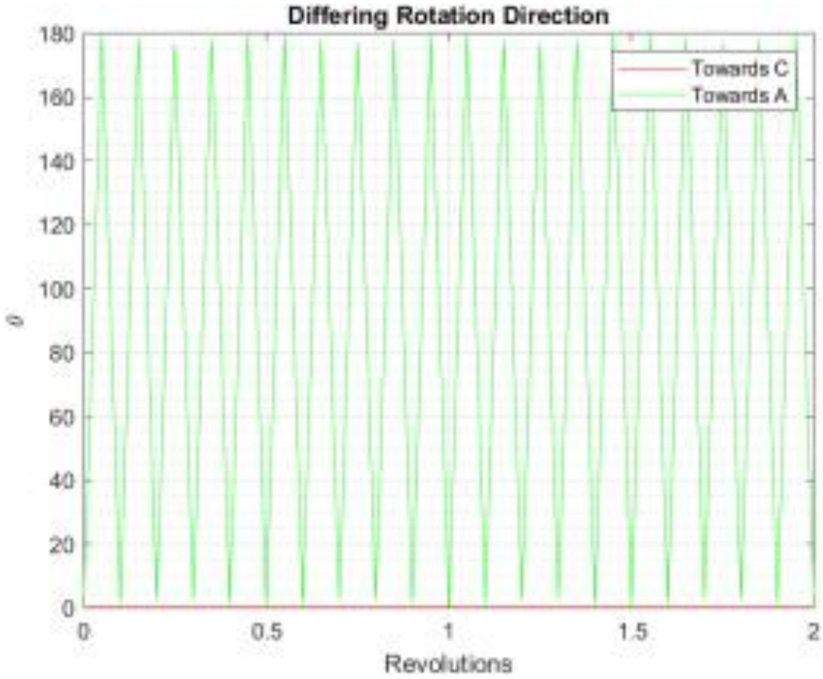

Fig. 7 Differing Rotation Direction. Note the constant stability towards the C direction.

Analytically the stability of the spacecraft can also be investigated using the linearized equations of motion for the spacecraft can be derived from Equations (24)-(30). The characteristic polynomial obtained provides insight into the spacecraft's stability both in the linear and nonlinear realm.

$$\delta\dot{q}_1 = 0 \qquad (50)$$

$$\delta\dot{q}_2 = \frac{\delta\omega_y}{2} + \delta q_3\Omega \qquad (51)$$

$$\delta\dot{q}_3 = \frac{\delta\omega_z}{2} + \delta q_2\Omega \qquad (52)$$

$$\delta\dot{q}_4 = 0 \qquad (53)$$

$$\delta\dot{\omega}_y = \theta\Omega\delta\omega_z \qquad (54)$$

$$\delta\dot{\omega}_z = -\Omega\delta\omega_y\mathrm{\theta} - 6\Omega^2\delta q_3 \qquad (55)$$

Table 2: Stability Characteristics

Spin about C axis. Spin = 5	Stability	Spin about C axis. Spin = 20	Stability
0.0000 + 14.9989i	Marginally Stable	0.0000 + 6.6655i	Marginally Stable
0.0000 - 14.9989i	Marginally Stable	0.0000 - 6.6655i	Marginally Stable
0.0000 + 0.0011i	Marginally Stable	-0.0000i + 0.0011i	Marginally Stable
0.0000 - 0.0011i	Marginally Stable	-0.0000 - 0.0011i	Marginally Stable

With a stable orientation for the spacecraft obtained, a small perturbation is applied in the nutation to further understand the spacecraft’s stability characteristics. In this case applying a larger perturbation results in less stability although the spacecraft is still marginally stable. Several different values were analyzed for the nutation perturbation and the spacecraft's spin rate. The resulting stability trends indicate that with increasing spin rate comes increased stability as shown in Figures (7)-(8), although the spacecraft is still marginally stable. Furthermore, the precession was unaffected with differing perturbations, a similar relationship was noticed with differing spin rates, although at a different rate.

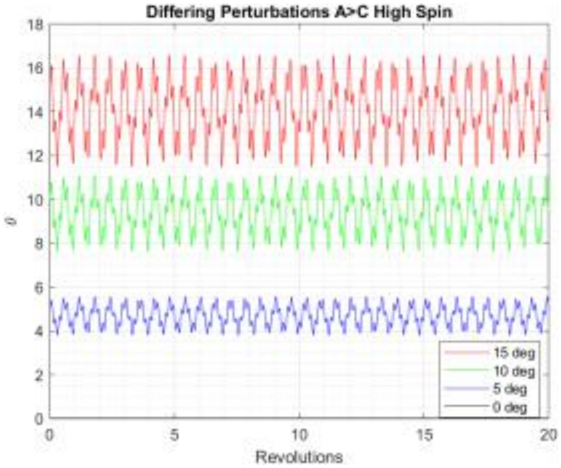

Fig. 8 Differing Perturbations A>C High Spin.

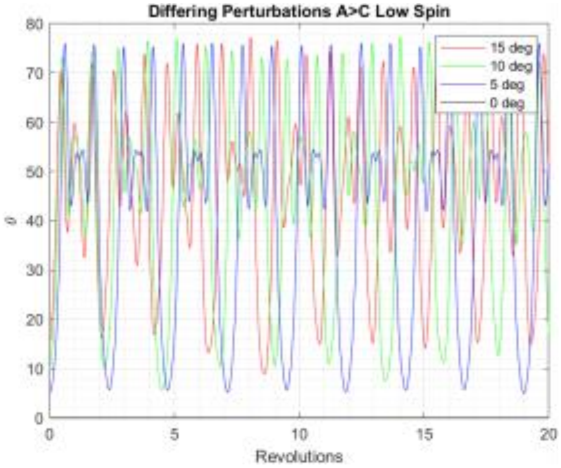

Fig. 9 Differing Perturbations A>C Low Spin.

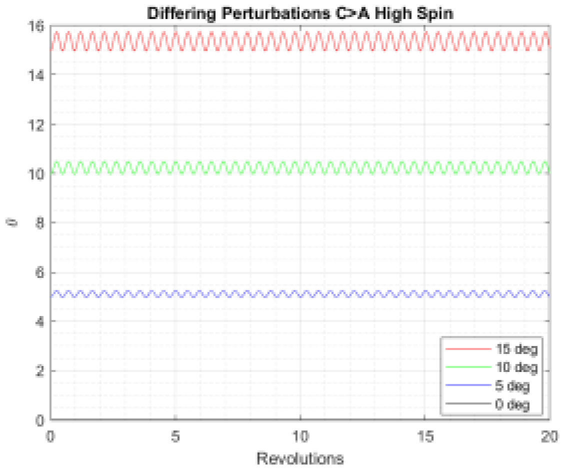

Fig. 10 Differing Perturbations C>A High Spin.

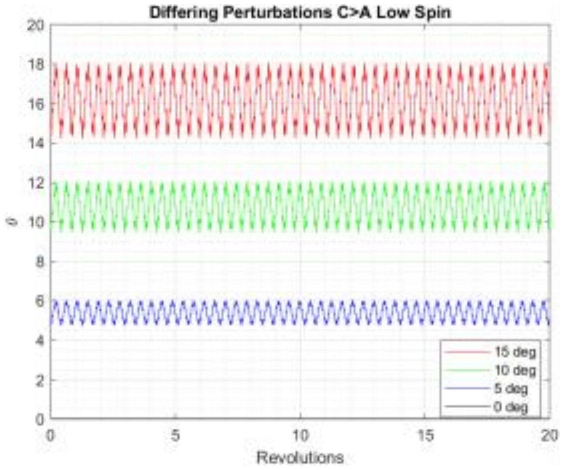

Fig. 11 Differing Perturbations C>A Low Spin.

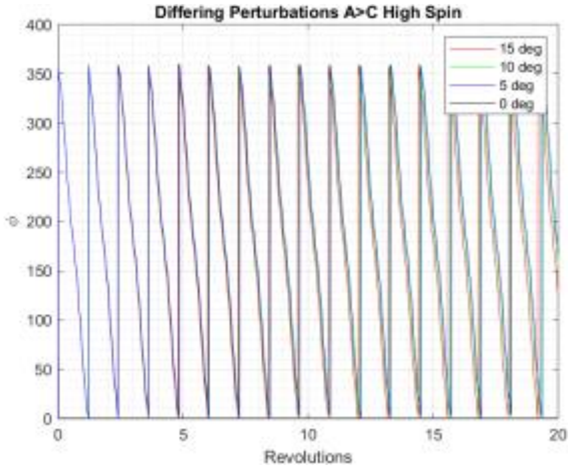

Fig. 12 Differing Perturbations A>C High Spin.

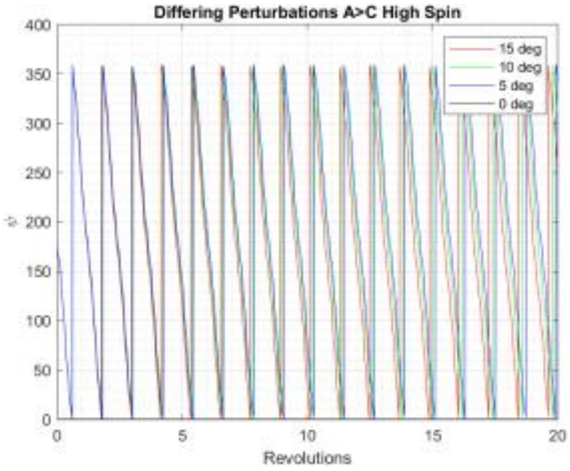

Fig. 13 Differing Perturbations A>C High Spin.

Lastly, the longer-term stability properties of the spacecraft under these conditions were also analyzed. By increasing the number of orbit revolutions and differing the orientations for both a stable and unstable orientation the results were unaffected.

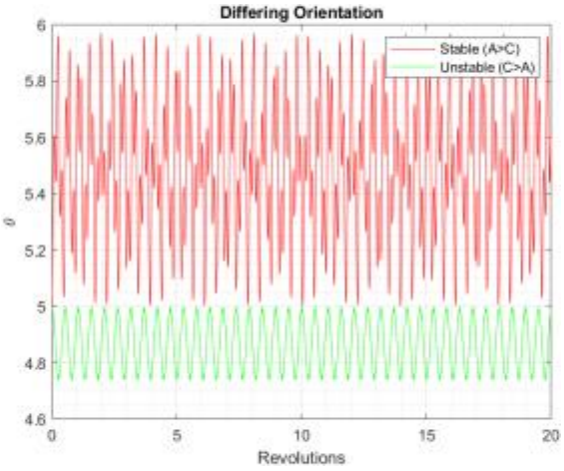

Fig. 14 Differing Orientations and Increased Revolutions.

E. Proposed Stability Method

For the most part, the cases represented here are already marginally stable. There are methods to make spacecraft stable, or in this case more stable. In this case, thrusters will be used to stabilize the spacecraft. Two thrusters will be placed across from each other and facing opposite directions about the symmetric axis. To prove that this is a reasonable method of stabilizing, ion thrusters will be used, which could add no more than 10 kg to the whole spacecraft. This makes them small enough that the thrusters plus their propellant would not change much. Equation (56) shows the change in angular momentum, or moment, created by thrusters being used:

$$M = \dot{H} = 2(r \times T\Delta t) \qquad (56)$$

Values can be selected for radius (r) as 1 meter and 0.05 N for thrust (T) as is reasonable for an ion thruster. Δt is the value that can be more freely changed, but in this case the thrusters will run for 5 orbits (a reasonable amount of time) which in this case amounts to 27,767 seconds since one orbital period at 400km orbital radius is 5,554 seconds. This leads to a provided torque (moment) of 2,777 Nms.

Since it is known that $H = I\omega$ then $\dot{H} = I\dot{\omega}$. Simplifying from here, a new equation of motion for the x-direction can be derived in Equation (57):

$$\dot{\omega}_x = \frac{2(r\times T\Delta t)}{C} \qquad (57)$$

Although thrusters are usually used to directly redirect a perturbed spacecraft, in this case they will cause a faster spin about the symmetric axis, which has already been known to stabilize the spacecraft. Making use of this method for a fraction of the full simulation time, in this case one fourth, will show the benefit of further spinning up the spacecraft via satellites. A comparison between w's and nutation for thrusters and no thrusters can be seen in Figures (15) and (16) for the already know stable case (C>A) starting at a low spin rate of 5 and nutation perturbation of 5 degree.

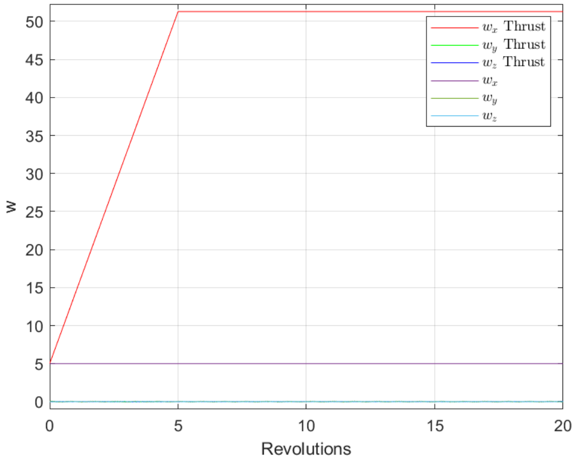

Fig. 15. W values with and without thrust (C>A)

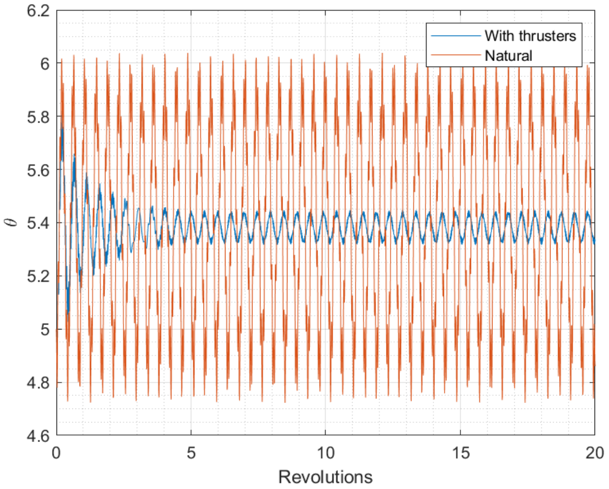

*Fig. 16**.** Nutation without and without thrusters (C>A).*

Nothing special happens with the nondimensionalized spins besides the increase in expected spin. The nutation can obviously be seen to still be marginally stable, although significantly more so than before, which could come in useful for precise needs of the spacecraft.

This can also be used in the unstable case (A>C) beneficially, although not to the same extent, as seen in nutation in Figure (16).

F. Conclusions

The variation of inertial properties for an arbitrary rigid body while applying torque-free perturbations leads to differing sinusoidal angular velocities over time. Several cases were plotted for this scenario. For a custom “C” shaped rigid body in orbit, it was found that the body was naturally stable, with k=1.98. Gravitational variation of the “C” shape spacecraft for several instances and orbits. While the spacecraft was already marginally stable, a proposed thruster control system was implemented for more control authority.

G. Appendix

MATLAB Code:

```
clc; clear; close all;

% Variables are repeated throughout to keep it simple since there would be
% an absurd amount of variables otherwise

global C A torque

% tolerance

obt = odeset('RelTol',1e-12,'AbsTol',1e-12);


%% A;C>B

% initial conditions
w1o = .1;               % rad/s
w2o = .1;               % rad/s
w3o = 5;                % rad/s
IC = [w1o w2o w3o];


% time step

t = 0:.01:100; % seconds


% run ode

[T,w1] = ode45(@eq1,t,IC,obt);


% plot omegas

figure(1);
plot(T,w1(:,1),'r',T,w1(:,2),'g',T,w1(:,3),'b');
ylabel('\omega (rad/s)'); xlabel('Time (sec)');
legend('$\omega_{x}$','$\omega_{y}$','$\omega_{z}$','Interpreter','latex');
title('A;C>B');


%% A;C>B
```

```
% initial conditions

w1o = .1;           % rad/s
w2o = .1;           % rad/s
w3o = 5;            % rad/s
IC = [w1o w2o w3o];

% time step

t = 0:.01:100;      % seconds

% run ode

[T,w2] = ode45(@eq2,t,IC,obt);


% plot omegas

figure(2);
plot(T,w2(:,1),'r',T,w2(:,2),'g',T,w2(:,3),'b');
ylabel('\omega (rad/s)'); xlabel('Time (sec)');
legend('$\omega_{x}$','$\omega_{y}$','$\omega_{z}$','Interpreter','latex');
title('A;C<B');


%% A<B<C

% initial conditions

w1o = .1;           % rad/s
w2o = 5;            % rad/s
w3o = .1;           % rad/s
IC = [w1o w2o w3o];

% time step

t = 0:.01:15;       % seconds


% run ode

[T,w3] = ode45(@eq3,t,IC,obt);


% plot omegas

figure(3);
plot(T,w3(:,1),'r',T,w3(:,2),'g',T,w3(:,3),'b');
ylabel('\omega (rad/s)'); xlabel('Time (sec)');
legend('$\omega_{x}$','$\omega_{y}$','$\omega_{z}$','Interpreter','latex');
title('A<B<C');


%% Custom Body

% initial conditions

w1o = .1;           % rad/s
w2o = .1;           % rad/s
w3o = 5;            % rad/s
IC = [w1o w2o w3o];


% time step

t = 0:.01:20;       % seconds


% run ode

[T,w4] = ode45(@eq4,t,IC,obt);
```

```
% plot omegas

figure(4);
plot(T,w4(:,1),'r',T,w4(:,2),'g',T,w4(:,3),'b');
ylabel('\omega (rad/s)'); xlabel('Time (sec)');
legend('$\omega_{x}$','$\omega_{y}$','$\omega_{z}$','Interpreter','latex');
title('Custom Rigid Body');


% define stability (k value)

A = 32464;
B = 19350;
C = 16264;
k = (A-C)*(B-C)/(A*B)*w3o^2;


% print k value

fprintf('K value of %.4f\n',k);


%% Gravity Gradient Task 6

% defined constants

C = 300;                % kgm^2 (changing variable)
A = 100;                % kgm^2 (changing variable)

mu = 398600;            % km^3/s^2
r = 6778;               % km

Omega = sqrt(mu/r^3);


% this next part is for plotting multiple runs at once using the command window

%figure(15);plot(T,a,'r',T,b,'g',T,c,'b',T,d,'k');xlabel('Revolutions');ylabel('\theta');legend('
15 deg','10 deg','5 deg','0 deg');title('Differing Perturbations');grid on;grid minor;


% initial conditions

thPert = 5;             % degrees

w1o = 5;                % non-dimensionalized
w2o = 0;                % non-dimensionalized
w3o = 0;                % non-dimensionalized
q1o = 0;
q2o = 0;
q3o = sind(thPert/2);   % (for perturbed linearized)
q4o = cosd(thPert/2);   % (for perturbed linearized)
% q3o = 0;              % (for unperturbed unlinearized)
% q4o = 1               % (for unperturbed unlinearized)

IC = [w1o w2o w3o q1o q2o q3o q4o];


% time step

t = 0:.01:20;           % revolutions


% run ode and seperate values

[T,qw6] = ode45(@eq6,t,IC,obt);
w6 = qw6(:,1:3);
q6 = qw6(:,4:7);
```

```matlab
% calculate angles

C11 = 1-2*q6(:,2).^2-2*q6(:,3).^2;
C21 = 2*(q6(:,1).*q6(:,2)+q6(:,3).*q6(:,4));
C31 = 2*(q6(:,3).*q6(:,1)-q6(:,2).*q6(:,4));
C13 = 2*(q6(:,1).*q6(:,3)+q6(:,2).*q6(:,4));
C12 = 2*(q6(:,2).*q6(:,1)-q6(:,3).*q6(:,4));

prec = wrapTo360(atan2d(C31,C21));
nut = acosd(C11);
spin = wrapTo360(atan2d(C13,C12));


% plot angles and w

figure(5);
plot(T,w6(:,1),'r',T,w6(:,2),'g',T,w6(:,3),'b');
ylabel('w'); xlabel('Revolutions');
legend('$w_{x}$','$w_{y}$','$w_{z}$','Interpreter','latex');
title('Gravity Gradient Omegas');

figure(6);
plot(T,nut);
ylabel('\theta'); xlabel('Revolutions');
title('Gravity Gradient Nutation');

figure(7);
plot(T,prec);
ylabel('\phi'); xlabel('Revolutions');
title('Gravity Gradient Precession');

figure(8);
plot(T,spin);
ylabel('\psi'); xlabel('Revolutions');
title('Gravity Gradient Spin');


% solving eigenvalues

x = C/A-1;          % shape factor
s = w1o - Omega;    % spin factor
y = w1o/Omega - 1;
Q = -(y+x*(1+y));

eigen = [0 -Omega 0 0;Omega 0 0 -6*Omega^2*x;.5 0 0 Q*Omega;0 .5 -Q*Omega 0];
L = eig(eigen);


%% Extra Credit (Thrusters)

% defined constants

period = 2*pi*sqrt(r^3/mu);             % seconds

thrust = .05;                           % N (this can be changed)
r = 1;                                  % m


% time step for thrusting and then no thrusting

to = 0:.01:5;                          % revolutions (only change final value)
t = to(length(to)):.01:t(length(t));    % revolutions
tspan = [to, t];

dt = period*to(length(to));             % seconds


% applied moment

torque = 2*(r*thrust*dt);               % Nms
```

```matlab
% run ode with thrusters (IC are from task 6)

[TT,qwto] = ode45(@eqt,to,IC,obt);


% run remaining ode without thrusters and separate values

IC = qwto(length(qwto),:);       % (new IC come from the end of last ode)
[TT,qwt] = ode45(@eq6,t,IC,obt);

QWT = [qwto;qwt];

wt = QWT(:,1:3);
qt = QWT(:,4:7);

% calculate nutation

C11 = 1-2*qt(:,2).^2-2*qt(:,3).^2;

nutT = acosd(C11);

% plot and compare w and nutation

figure(10);
plot(tspan,wt(:,1),'r',tspan,wt(:,2),'g',tspan,wt(:,3),'b');
hold on
plot(T,w6(:,1),'-',T,w6(:,2),'-',T,w6(:,3),'-');
ylabel('w'); xlabel('Revolutions');
legend('$w_{x}$ Thrust','$w_{y}$ Thrust','$w_{z}$
Thrust','$w_{x}$','$w_{y}$','$w_{z}$','Interpreter','latex');
title('Gravity Gradient Omegas with Aid');
grid on; ylim([-1 wt(length(wt),1)+1]);

figure(11);
plot(tspan,nutT,T,nut);
ylabel('\theta'); xlabel('Revolutions');
title('Gravity Gradient Nutation with Aid');
legend('With thrusters','Natural');
grid on; grid minor;


%% ode45

% A;C>B
function wdot = eq1(t,x)

A = 500;
B = 100;
C = 500;

% to output [w1 w2 w3]

wdot = zeros(3,1);
wdot(1) = -(C-B)*x(2)*(x(3))/A;
wdot(2) = -(A-C)*x(1)*(x(3))/B;
wdot(3) = -(B-A)*x(1)*x(2)/C;
end

% A;C<B
function wdot = eq2(t,x)

A = 100;
B = 500;
C = 100;

% to output [w1 w2 w3]

wdot = zeros(3,1);
wdot(1) = -(C-B)*x(2)*x(3)/A;
wdot(2) = -(A-C)*x(1)*x(3)/B;
wdot(3) = -(B-A)*x(1)*x(2)/C;
end


% A<B<C
function wdot = eq3(t,x)

A = 100;
B = 300;
C = 500;

% to output [w1 w2 w3]

wdot = zeros(3,1);
wdot(1) = -(C-B)*x(2)*x(3)/A;
wdot(2) = -(A-C)*x(1)*x(3)/B;
wdot(3) = -(B-A)*x(1)*x(2)/C;
end

% custom
function wdot = eq4(t,x)

A = 32464;
B = 19350;
C = 16264;

% to output [w1 w2 w3]

wdot = zeros(3,1);
wdot(1) = -(C-B)*x(2)*x(3)/A;
wdot(2) = -(A-C)*x(1)*x(3)/B;
wdot(3) = -(B-A)*x(1)*x(2)/C;
end

% gravity gradient
function dot = eq6(t,x)

global C A


s = x(1) - 1;    % spin factor

% C is symmetry axis

sh = C/A-1;      % shape factor

% to output [w1 w2 w3 q1 q2 q3 q4 w2thrust]

dot = zeros(7,1);
dot(1) = 0;
dot(2) = 2*pi*(-x(3)*s+sh*(-x(3)*x(1)+12*(x(5)*x(6)-x(4)*x(7))*(x(4)*x(5)+x(6)*x(7))));
dot(3) = 2*pi*(x(2)*s+sh*(x(1)*x(2)-6*(1-2*x(4)^2-2*x(6)^2)*(x(4)*x(5)+x(6)*x(7))));
dot(4) = (pi) * (x(5)*x(3) - x(6)*x(2) + x(7) * (x(1) - s - 1));
dot(5) = (pi) * (x(7)*x(2) - x(4)*x(3) + x(6) * (x(1) - s + 1));
dot(6) = (pi) * (x(4)*x(2) + x(7)*x(3) - x(5) * (x(1) - s + 1));
dot(7) = -(pi) * (x(5)*x(2) + x(6)*x(3) + x(4) * (x(1) - s - 1));
end

% extra credit
function dot = eqt(t,x)
global C A torque

s = x(1) - 1;    % spin factor

% C is symmetry axis

sh = C/A-1;      % shape factor


% to output [w1 w2 w3 q1 q2 q3 q4]
```

```
dot = zeros(7,1);
dot(1) = torque/C;
dot(2) = 2*pi*(-x(3)*s+sh*(-x(3)*x(1)+12*(x(5)*x(6)-x(4)*x(7))*(x(4)*x(5)+x(6)*x(7))));
dot(3) = 2*pi*(x(2)*s+sh*(x(1)*x(2)-6*(1-2*x(4)^2-2*x(6)^2)*(x(4)*x(5)+x(6)*x(7))));
dot(4) = (pi) * (x(5)*x(3) - x(6)*x(2) + x(7) * (x(1) - s - 1));
dot(5) = (pi) * (x(7)*x(2) - x(4)*x(3) + x(6) * (x(1) - s + 1));
dot(6) = (pi) * (x(4)*x(2) + x(7)*x(3) - x(5) * (x(1) - s + 1));
dot(7) = -(pi) * (x(5)*x(2) + x(6)*x(3) + x(4) * (x(1) - s - 1));
end
```

H. References

"Ion Thruster," *Wikipedia* Available: https://en.wikipedia.org/wiki/Ion_thruster.